Claim 1-2 and 5-7 stand rejected under 35 U.S.C. 102(b) as being anticipated by

Dale et al. (U.S. Patent No 6,272,673, issue date August 7, 2001). Applicants respectfully

traverse.

35 U.S.C. 102(b) states:

A person shall be entitled to a patent unless –
    (b) the invention was patented or described in a printed publication in this or a
foreign country or in public use or on sale in this country, more than one year
prior to the date of the application for patent in the United States...

The issue date of the Dale patent is August 7, 2001, which is after the filing date

of the present application (December 3, 1998) and not "more than one year prior to the

date of the application." Therefore, it is respectfully submitted that the rejection of

claims 1-2 and 5-7 under 35 U.S.C. 102(b) is improper and the rejection is understood to

be under 35 U.S.C. 102(e).

In addition, claim 1 recites a system for transporting objects between a first and

second machine where the first machine is programmed in a first language and the second

machine is programmed in a second language, the system comprising, among other

things, a first processor on the first machine for executing code and instantiating an object

on the first machine and an output for outputting the object with persistence information

to the second machine wherein after the object is output from the first machine, a first

processor deletes the instantiation of the object from the first machine. Dale does not

teach each limitation of the claim 1 invention. Thus, it is respectfully submitted that Dale

does not teach each and every aspect of the claim 1 invention and the rejection should be

withdrawn.

The disclosure in the Dale reference that the Examiner cites at col. 6, lines 24-25, teaches applets that "may alternatively be implemented in another object-oriented language." "Applets" do not equate to a first machine programmed in a first language and a second machine programmed in a second language as recited in the claim 1 invention. The "applets" as disclosed by Dale, are components that are executable code that is embodied as JAVA classes. Dale discloses that the *applets* may "alternatively be implemented in another object-oriented language." Notably, it is the *applets* that may be implemented in another object-oriented language. Thus, contrary to the Examiner's assertions, Dale merely discloses that an applet may be encoded in JAVA or another object-oriented language but does not teach or suggest a first machine programmed in a first language and a second machine programmed in a second language. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 1 invention and the rejection should be withdrawn.

In addition, contrary to the Examiner's assertions, Dale does not teach a first processor on a first machine for executing code and instantiating an object on the first machine and an <u>output for outputting the object with persistence information to the second machine</u>. The Examiner relies on Dale, col. 12, lines 21-22 (and Fig. 8B) to support this rejection. However, this reliance is misplaced. Dale discloses a component 64 to be instantiated on an application server 24a but the application server 24a does not contain an output for outputting the object with persistence information to a second machine. As seen in Fig. 8B, component 64 is instantiated on the application server 24a (col. 12, lines 19-21) but there is no teaching or suggestion to output the object (i.e. component 64) with persistence information to a second machine as recited. Also,

4

component 63 is instantiated on client 20a but outputting the object (i.e. component 63) with persistence information to a second machine is also not disclosed. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 1 invention and the rejection should be withdrawn.

The disclosure of Dale cited by the Examiner at column 12, lines 17-18, "application server 24a provides the HTML page 62 to the client 20a" does not anticipate an output for outputting the object with persistence information to a second machine as recited in the claim 1 invention. The Examiner appears to attempt to equate the HTML page 62 of Dale with the object of the present invention. However, the HTML page 62 of Dale is not the same as the object of the present invention at least because there is no disclosure of a first processor on the first machine (the Examiner apparently equates the application server 24a with the first machine) and instantiating the object on the first machine/application server 24a. Instead, the application server 24a of Dale "provides the HTML page 62 to the client 20a." There is no teaching or suggestion that a processor on the application server 24a instantiates the HTML page 62 on the application server 24a (i.e., the "first machine"). Moreover, there is no teaching of outputting the object with persistence information as recited in the claim 1 invention. Dale does not teach or suggest persistence information at all. The Examiner seems to assert that "the object with persistence information" (in the present application) is supposedly the same as "application server 24a provides the HTML page 62 to the client." (col.12, lines 17-18 of Dale). On the contrary, these are not equivalent because 1) there is no teaching of persistence information and 2) there is no first processor for instantiating the object on the first machine. Dale does not teach persistence information in this passage. It is

5

respectfully requested that the Examiner provide a specific cite in the reference where persistence information is taught if the Examiner maintains this rejection on this ground. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 1 invention and the rejection should be withdrawn.

The disclosure of Dale at column 13, line 55 cited by the Examiner where "the component is explicitly destroyed" is not the same as "after said object is output from the first machine, the first processor deletes the instantiation of the object from said first machine" as recited in the claim 1 invention. Dale refers to application components connected through functional connections (col. 12, lines 35-40). A list of registrations of application components is maintained to establish connections (col. 12, lines 41-43). Dale further discloses that a component may become "unregistered" such as occurs when the component is no longer instantiated, such as if the component is explicitly destroyed (col. 13, lines 52-55). Dale merely suggests deleting an object however Dale does not teach or suggest *outputting an object* with persistence information *to a second machine* and *after the object is output* from the first machine, *deleting the instantiation of the object from the first machine*. Thus, even if one were to assume the "component" of Dale is the "object" of the present invention, Dale still does not teach *after the object is output from the first machine* deleting the instantiation of the object from the first machine. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 1 invention and the rejection should be withdrawn.

Claim 2 recites, among other things, the limitations of claim 1 and a second processor on the second machine for receiving the object with persistence information

and allowing interaction with said object, the interaction creating events. Dale does not teach each and every aspect of the claim 2 invention.

The disclosure in Dale as cited by the Examiner at column 10, line 18, "the client 20a receives a requested HTML page" does not anticipate the limitation "…for receiving the object with persistence information…" as purported by the Examiner. Nowhere does Dale teach or even suggest the object with persistence information. Merely stating receiving a requested HTML page does not teach the object with persistence information. Also, merely stating "receiving a requested HTML page" does not teach allowing interaction with the object, said interaction creating *events*. There is no mention of events in the cited Dale disclosure. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 2 invention and the rejection should be withdrawn.

The disclosure in Dale as cited by the Examiner at col. 10, lines 18-20, "…if the client 20a detects an applet tag for a component, the browser instantiates the component" does not anticipate the limitation "…and allowing interaction with the object, the interaction creating events" as suggested by the Examiner. The disclosure in Dale cited by the Examiner merely refers to instantiating the component in the client 20a. Notably, Dale does not teach "allowing interaction with the object" and further does not teach the aspects of the claim 2 invention discussed above. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 2 invention and the rejection should be withdrawn.

The two-way communication arrows depicted in Fig. 7 and column 10, lines 30-49 does not anticipate "the interaction creating events" as cited in the claim 2 invention as the Examiner asserts. The passage in Dale refers to maintaining a registration of

components associated with applications and handling requests from various components for connection to other components. Nowhere in the passage does Dale teach or even suggest creating events. Nor does Fig. 7 provide any additional support for the Examiner's contention. Double-headed arrows illustrated in Fig. 7 does not teach creating events and it is not clear on what basis the Examiner has determined that they do. One of skill in the art would know that double-headed arrows do not imply creating events. In fact, double-headed arrows bear no relationship to events at all. Moreover, double-headed arrows do not imply allowing interaction with the object, the interaction creating events, specifically. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 2 invention and the rejection should be withdrawn.

Claim 5 recites a data structure comprising, among other things, persistence information associated with a first object representation and event information relating to interaction with the object. Dale does not teach each and every aspect of the claim 5 invention.

The applet tag that the Examiner cites at column 10, line 19 of the Dale reference does not anticipate persistence information of the claim 5 invention as the Examiner implies because an applet tag is not equivalent to persistence information. Applets as defined in the Dale reference are "small portions of executable code that can be incorporated into HTML pages..." (col. 6, lines 20-22). Persistence information refers to information for which a data object exists. One of skill in the art would not equate these two concepts because they are not related to one another. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 5 invention and the rejection should be withdrawn.

Fig. 6, steps 604 and 605 in the Dale reference does not anticipate event information relating to interaction with the object as recited in the claim 5 invention. Step 604 in Fig. 6 refers to registering a component's existence via an infrastructure. Step 605 refers to a browser rendering a page if no applet tag is found for the component. Not only do neither steps 604 nor 605 teach an interaction with an object, but they also do not teach a data structure comprising event information. It is unclear how the Examiner has equated registering a component's existence and rendering a page with no applet tag (disclosed in Dale) with a data structure comprising event information relating to interaction with an object (claim 5 invention) because the concepts are completely dissimilar and unrelated to each other. Thus, it is respectfully submitted that Dale does not teach each and every aspect of the claim 5 invention and the rejection should be withdrawn.

Claim 6 stands rejected for the same reasons set forth in the Office Action dated March 14, 2002 for claim 1 and claim 7 stands rejected for the same reasons set forth in the Office Action dated March 14, 2002 for claim 2. It is respectfully submitted that both claim 6 and claim 7 are allowable for at least the reasons set forth for claims 1 and 2, respectively.

Claims 3 and 8 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Dale in view of Chang (U.S. Patent No. 5,960,436). Applicants respectfully traverse.

Claim 3 recites, among other things, an output of the second machine for outputting the events and objects with the persistence information to the first machine wherein the first machine reinstantiates the objects based on the persistence information and handles the events as effecting the reinstantiated objects.

The Examiner admits that Dale does not teach or suggest these limitations but relies on Chang to make up for the deficits. However, Chang does not cure the deficiencies of Dale.

The disclosure of Chang cited by the Examiner at column 2, lines 8-9, "...the client's local file system which dynamically creates a record of the transaction and of the modified objects..." is not equivalent to an output for outputting events and objects with persistence information to a first machine as recited in the claim 3 invention. In Chang's disclosure, there is no teaching or suggestion of outputting events. There is no mention of events at all. The Examiner seems to be attempting to equate the "record of the transaction" of Chang with the "events" of the present invention. This is an improper coupling and is at best an exercise in improper hindsight. The "record of the transaction" is completely unrelated to "events" and one of ordinary skill in the art would not equate the two concepts. The "record of the transaction" is merely a record of the local transaction performed at the client. It is not itself an event and, further, is not suggested or disclosed as being handled in the first machine as effecting reinstantiated objects.

Moreover, Chang states that "...when the client subsequently reconnects with the server, the logged transactions are read from the client's local file system, sent to the server, and replayed on the server, with the modified objects being written back to the server's database" but does not teach or suggest persistence information as recited in the claim 3 invention. The Examiner attempts to equate the logged transactions as disclosed by Chang with the persistence information recited in the claim 3 invention. Notably, the Examiner appears to alternately refer to the transactions as "events" on the one hand and then as "persistence information" on the other hand. Actually, the transactions as

10

disclosed by Chang are not the same as either the events or the persistence information. One of ordinary skill in the art would understand that events, persistence information and "transactions" as disclosed by Chang are not the same as explained above. Indeed, Chang does not teach or suggest events or persistence information at all. Thus, it is respectfully submitted that the Examiner has not established a prima facie case of obviousness and the rejection should be withdrawn.

Claim 8 was rejected for the same reasons as claim 3. Because claim 3 is allowable, it is respectfully submitted that the rejection of claim 8 should also be withdrawn.

Claims 4 and 9 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Dale in view of Barlow (U.S. Patent Number 6,275,935) and Chang.

Claim 4 recites, among other things, an input in a first machine for receiving persistence information and an event from a second machine, a processor in the first machine for instantiating an object based in part on the persistence information and an event handler in the first machine for handling the event in combination with modifying the object and an output for outputting the modified object to a second machine.

Dale does not teach or suggest an input in the first machine for receiving persistence information and an event. The Examiner now equates the client 20a in Figure 3 of Dale with the first machine. However, the client 20a in Figure 3 of Dale does not receive persistence information and an event from a second machine. Thus, the client 20a in Figure 3 of Dale cannot be equivalent to the first machine. The Examiner equates an HTML page at column 10, line 18 of Dale with the persistence information and the event of the claim 4 invention. However, the HTML page is not the same and is not suggestive

of persistence information. An HTML page is merely a web page whereas persistence

information is information for which a permanent data object exists. Nor is the HTML

page the same or suggestive of an event. An HTML page is merely a web page and does

not relate to an occurrence or happening of significance to a task or program, for

example. Also, the client 20a in Figure 3 of Dale is not disclosed as having an output for

outputting a *modified* object to a second machine.

Dale does not teach or suggest a processor in the first machine for instantiating an

object based in part on the persistence information as the Examiner asserts. The Examiner

has equated the client 20a in Figure 3 of Dale with the first machine recited in the claim 4

invention. However, the client 20a in Figure 3 of Dale does not comprise a processor for

instantiating an object based in part on the persistence information. The Examiner

appears to equate an applet tag disclosed by Dale with the persistence information recited

in the claim 4 invention. It is respectfully pointed out that an applet tag is not related to

and is not suggestive of persistence information. An applet tag as define by Dale is "small

portions of executable code that can be incorporated into HTML pages." (col. 6, lines 19-

21). This is clearly not related to persistence information and one of ordinary skill in the

art would not confuse the two different entities. However, even if one were to erroneously

conclude that applet tags were somehow related to persistence information, Dale still

does not teach or suggest instantiating an object based in part on the applet tags.

Chang does not cure the deficiencies of Dale for reasons set forth above. Barlow,

either alone or in combination with Chang, likewise does not cure the deficiencies of

Dale. Neither Barlow, Chang, nor Dale, either alone or in combination, teach or suggest

receiving persistence information and an event from a second machine and instantiating
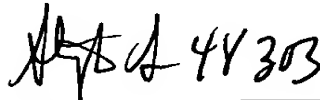
an object based in part on the persistence information. The Examiner asserts that Barlow discloses an event handler at col. 17, lines 39-40 and modifying an object at col. 17, line 40 and concludes that one of ordinary skill in the art would somehow be motivated to perform the claim 4 invention based on these disclosures. It is respectfully submitted that there would be no motivation to perform the claim 4 invention based on these disclosures. Contrary to the Examiner's assertions, Barlow does not teach or suggest modifying the object. The term "executed" at col. 17, line 40 on which the Examiner relies to provide this teaching actually refers to executing the event and does not refer to modification of objects. Thus, Barlow does not teach or suggest an event handler in the first machine for handling the event *in combination with modifying the object*. Thus, it is respectfully submitted that the Examiner has not established a prima facie case of obviousness and the rejection should be withdrawn.

To establish a *prima facie* case of obviousness, there must be some suggestion or motivation to one of ordinary skill in the art to modify the references or to combine reference teachings. In this case, none of the cited references suggests the claimed invention. Indeed, none of the cited references teaches or suggests receiving persistence information and an event, instantiating an object based in part on the persistence information, handling the event in combination with modifying the object and outputting the modified object to a second machine. Also, all claim limitations must be taught or suggested by the prior art to establish a *prima facie* case of obviousness. In this case, none of the limitations set forth above is taught or suggested by any of the cited references either alone or in combination. Therefore, it is respectfully submitted that the claim 4 invention is allowable over the cited references and the rejection should be withdrawn.

Claim 9 was rejected for the same reasons as claim 4. Therefore, claim 9 should be allowable for at least the reasons set forth for claim 4 above.

Applicants respectfully submit that the instant application is in condition for allowance. If the Examiner feels, however, that further amendment and/or discussion may be helpful in facilitating prosecution of the case, the Examiner is respectfully requested to telephone the undersigned attorney of record at the number appearing below.

Respectfully submitted,

Christopher R. Glembocki
Registration No. 38,800

BANNER & WITCOFF, LTD.
1001 G Street, N.W.
Washington, D.C. 20001
(202) 508-9100

Date: May 23, 2002

MARKED-UP VERSION OF THE AMENDMENTS

In the Specification:

Please amend the paragraph at page 1, lines 8-13 as follows:

--The present application is related to U.S. Serial No. 08/959,300, entitled "Method and Apparatus for Automatic Generation of Text and Computer-Executable Code", filed on October 28, 1997. Also, the present application is related to U.S. Serial No. 09/223,773, "A System for Converting Event-Driven Code into Serially Executed Code", filed on December 31, 1998; to U.S. Serial No. 09/223,565, entitled "Drag and Drop Creation and Editing of A Page Incorporating Scripts", filed on December 31, 1998; and, to U.S. Serial No. 09/223,774, entitled "Page Object Model", filed on December 31, 1998.--

In the Claims

Please amend claim 3 as follows:

3. (Amended) The system according to claim [1] 2, further comprising;

an output of said second machine for outputting said events and said objects with said persistence information to said first machine,

wherein said first machine reinstantiates said objects based on said persistence information and handles said events as effecting said reinstantiated objects.